



Public Cloud

Deploying the Security Fabric as Infrastructure as
Code in Microsoft Azure

Joeri Van Hoof

@jvhoof

What is Infrastructure as Code all about?

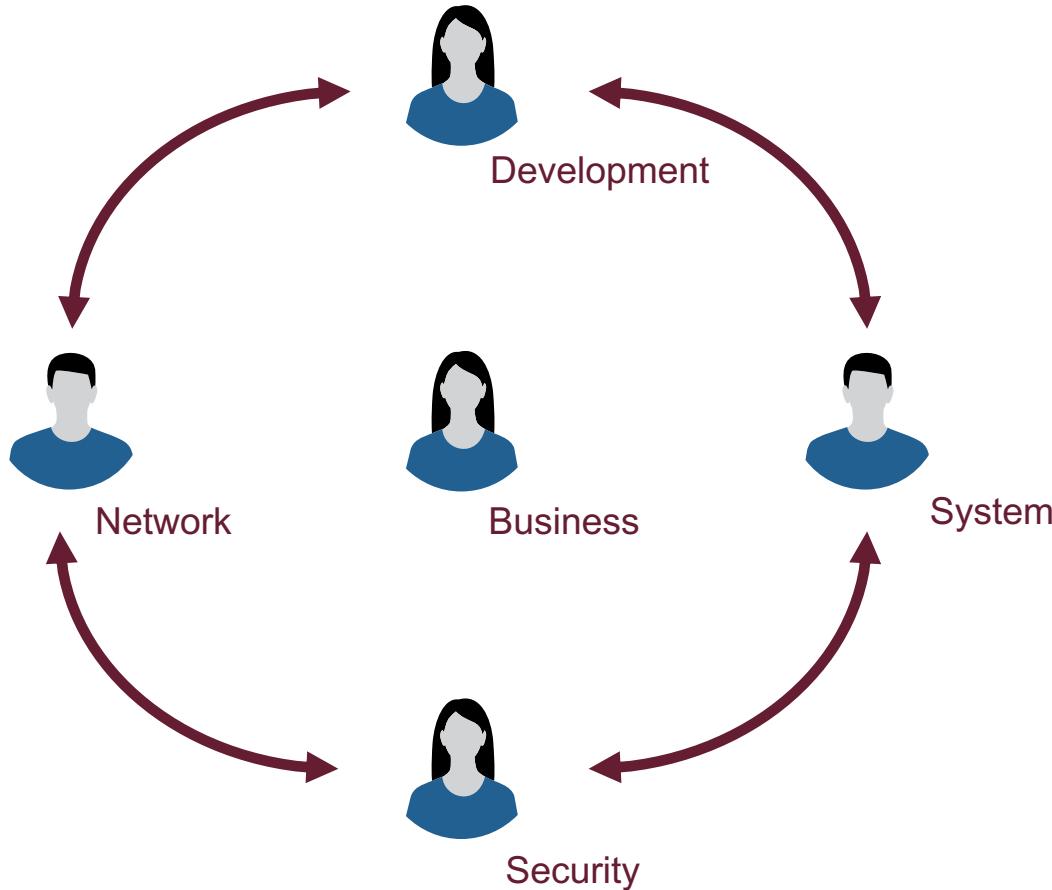
A little story to start...



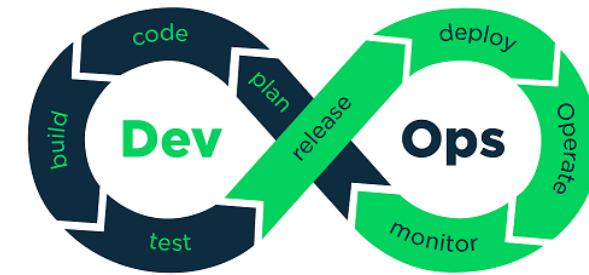
Asterix and Obelix - The 12 tasks

<https://www.youtube.com/watch?v=JOhRhq6Pr6g>
(watch around minute 40 and on)

A little story to start...



DevOps



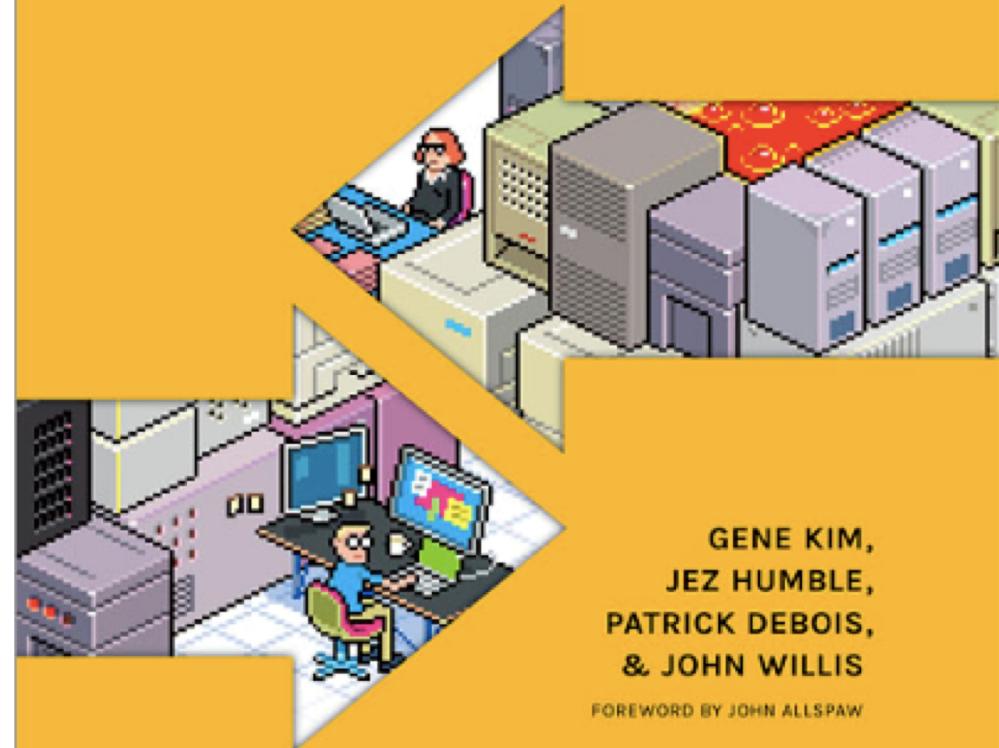
What is DevOps?

- “DevOps and its resulting technology, architectural, and cultural practices represent a convergence of many philosophical and management movements....
- ...DevOps is the outcome of applying the most trusted principles from the domain of physical manufacturing and leadership to the IT value stream DevOps relies on bodies of knowledge from **Lean**, **Theory of Constraints**, the **Toyota Production System**, **reliance engineering**, **learning organizations**, **safety culture**, **human factors**, and many others.”

• ~ *The DevOps Handbook*

The DevOps Handbook

HOW TO CREATE WORLD-CLASS
AGILITY, RELIABILITY, & SECURITY
IN TECHNOLOGY ORGANIZATIONS



The Phoenix Project

"The Phoenix Project" describes the underpinning principles that all the DevOps patterns can be derived from '**The Three Ways**'.

- **First Way:** left-to-right flow of work from Development to IT Operations to the customer.
- **Second Way:** constant flow of fast feedback from right-to-left at all stages of the value stream,
- **Third Way:** creating a culture that fosters continual experimentation, identify repetition and practice.



The Phoenix Project

A Novel About IT, DevOps,
and Helping Your Business Win

REVISED
WITH NEW
RESOURCE
GUIDE

Gene Kim, Kevin Behr, and George Spafford

DevOps

Development vs. Operation

How can we make this cooperation smoother?

Started with Devops Days in Ghent 2009



Infrastructure as Code

Infrastructure as Code

What?

The essence of IaC is to treat the configuration and definition of systems just like writing software

Infrastructure as Code

Why?

- Repeatability
- Speed
- Accuracy
- Testability
- Agility

Infrastructure as Code

Advantages

- Not just for cloud, software defined data center
- Embedded documentation
- Source control
- Flexible build process
- Removes or reduces configuration drift

Infrastructure as Code

| People | Process | Products |
|--|--|---|
| <ul style="list-style-type: none">• Required to be successful• A way of life• Connects into the DevOps movement• IaC / DevOps movement• Grassroots vs top down | <ul style="list-style-type: none">• Simplicity• Modular• Flexible• Versioning | <ul style="list-style-type: none">• Terraform, Ansible, Salt, Chef, Puppet• ARM templates• Powershell / Bash• Azure Quickstart templates• Visual Studio Code• CI/CD pipeline tools like:<ul style="list-style-type: none">• Azure Devops• Jenkins• Circle CI• ... |

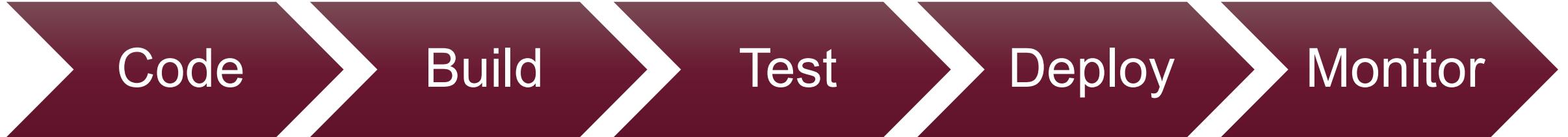
Infrastructure as Code

People: your first steps

- Find something small to automate
 - Fail fast, fail often!
- Set your and the organisations expectations
 - Don't just fall for buzzwords!
- Verify that it works
 - Did this help in our deployment efforts?
- Engage your peers
 - Share your experience
- Rinse and repeat to refine the process

Infrastructure as Code

Process



Microsoft
OMS



CI
Continuous Integration

CD
Continuous Deployment

Before the products

some extra concepts that are important...

Configuration management

“Configuration management is the process of **standardizing resource configurations** and **enforcing** their **state** across IT infrastructure in an automated yet agile manner.”

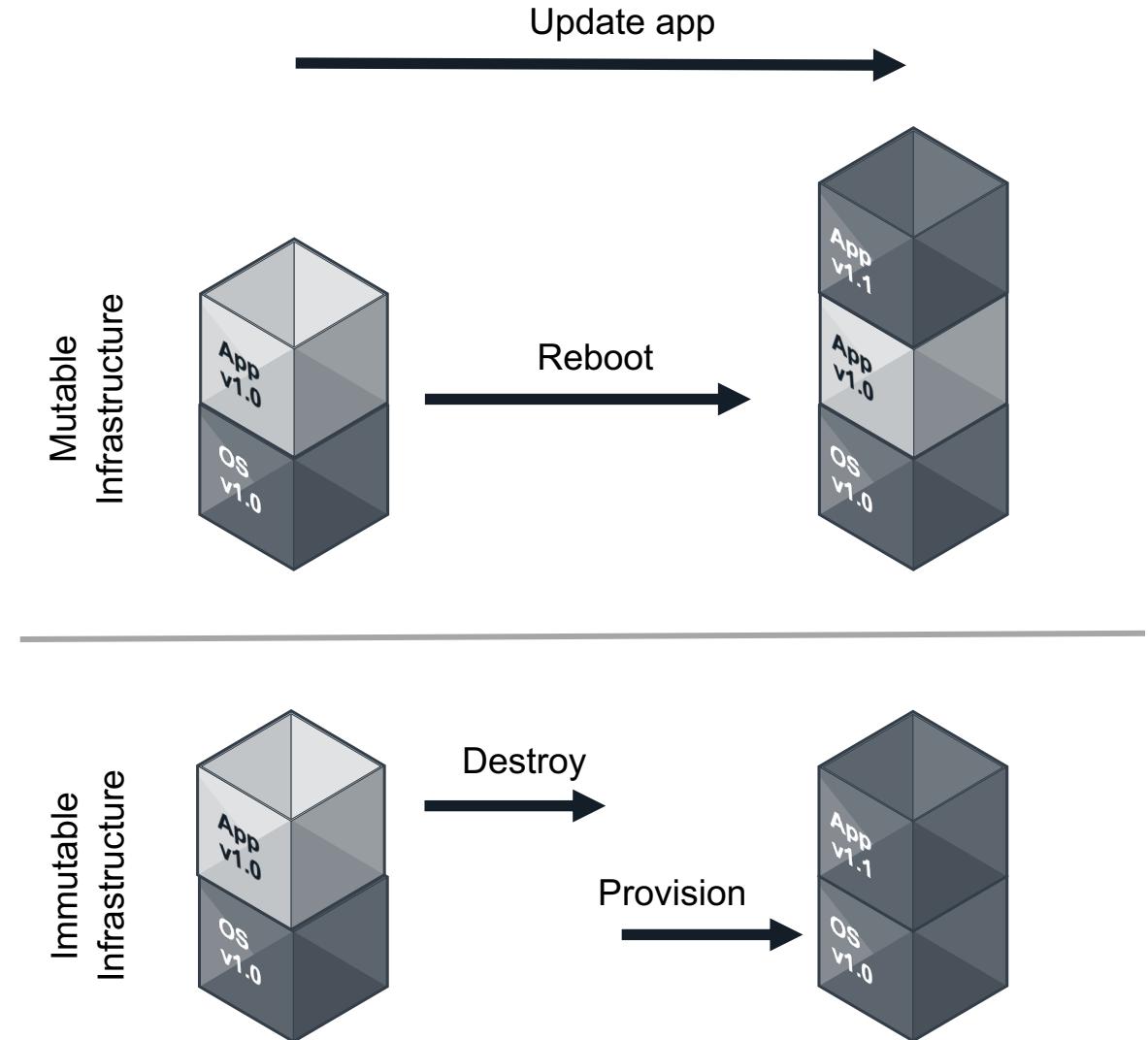
Puppet Labs

Idempotence

Idempotence is the property that a deployment command always sets the **target environment** into the **same configuration, regardless** of the environment's **starting state**.

Mutable vs. Immutable

- Immutable means deploy will never change
- Image regenerated each time
- Reuse the image from dev to QA to Prod
- To scale spin up an new instance
- Reliability, predictability



Imperative vs Declarative

- Imperative (procedural): defines specific commands that need to be executed in the appropriate order to arrive at the end state
- Aka “the HOW”

```
# AZURE CLI:  
# Create resource group  
  
rg="myResourceGroup"  
Location="westeurope"  
  
echo "--> Creating $rg resource  
az group create --location "$location" --name "$rg"
```

Imperative vs Declarative

- Declarative (functional): defines the desired state and the system executes what needs to happen to achieve that desired state
- Aka “the WHAT”

```
# Azure ARM Template
...
"resources": [
    {
        "type": "Microsoft.Resources/resourceGroups",
        "apiVersion": "2018-05-01",
        "location": "[parameters('rgLocation')]",
        "name": "[parameters('rgName')]",
        "properties": {}
    }
]
...
...
```

Products overview

| Tool | Tool type | Infrastructure | Architecture | Approach | Language |
|--|--------------------------|----------------|--------------|--------------------------|------------|
|  puppet | Configuration Management | Mutable | Pull | Declarative | DSL + Ruby |
|  CHEF | Configuration Management | Mutable | Pull | Declarative & Imperative | Ruby |
|  ANSIBLE | Configuration Management | Mutable | Push | Declarative & Imperative | YAML |
|  SALTSTACK | Configuration Management | Mutable | Push & Pull | Declarative & Imperative | YAML |
|  Terraform | Provisioning | Immutable | Push | Declarative | HCL |

Products overview

| Tool | Tool type | Infrastructure | Architecture | Approach | Language |
|---|--------------|----------------|--------------|-------------|--------------|
| AWS CloudFormation Templates | Provisioning | Immutable | Push | Declarative | JSON |
| | | | | | |
| ARM Templates | Provisioning | Immutable | Push | Declarative | JSON or YAML |
| | | | | | |
|  HashiCorp Terraform | Provisioning | Immutable | Push | Declarative | HCL |
| | | | | | |

So many choices!

What do we see in the field?

- Azure: ARM Templates
- AWS: CFT templates
- Multi-Cloud: Terraform + Ansible

So many choices!

During the workshop?

- Microsoft Azure
- Azure DevOps
- Azure Resource Manager (ARM) Templates
- Terraform
- Ansible
- Azure Cloud Shell

Infrastructure as Code

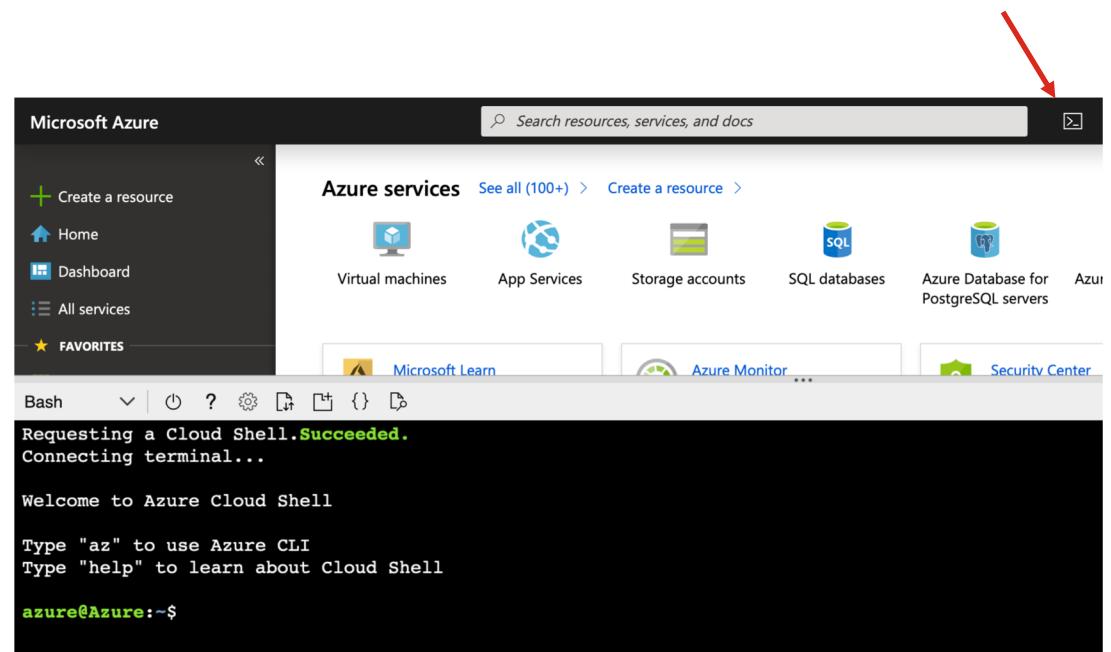
LAB 1: Our first baby steps

License terms validation

If you have never deployed a FortiGate or FortiAnalyzer in your subscription you need to accept the terms.

Either you deploy the product in Azure via the Azure Portal or you can use the Azure Cloud Shell and run the below commands:

```
$ az vm image accept-terms --publisher fortinet --offer fortinet_fortigate-vm_v5 --plan fortinet_fg-vm  
$ az vm image accept-terms --publisher fortinet --offer fortinet-fortianalyzer --plan fortinet-fortianalyzer
```



ARM template validation

What if you are developing an ARM templates? Once you commit to git is this template correct? Can we automate the testing in a repeatable, consistent way?

Azure DevOps

Plan smarter, collaborate better and ship faster with a set of modern dev services.

[Start free >](#)

 [Start free with GitHub >](#)

Already have an account?

[Sign in to Azure DevOps >](#)

ARM template validation

The ingredients

- Microsoft Azure account
- Microsoft Azure DevOps account (free):
 - <https://azure.microsoft.com/en-in/services/devops/>
 - <https://github.com/jvhoof/xperts-academy-2019/>
- Optional: Github account

Azure DevOps

Plan smarter, collaborate better and ship faster with a set of modern dev services.

[Start free >](#)

 [Start free with GitHub >](#)

Already have an account?

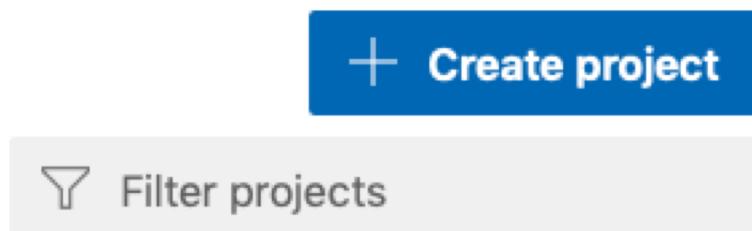
[Sign in to Azure DevOps >](#)

ARM template validation

Create a new project in the main organization page.

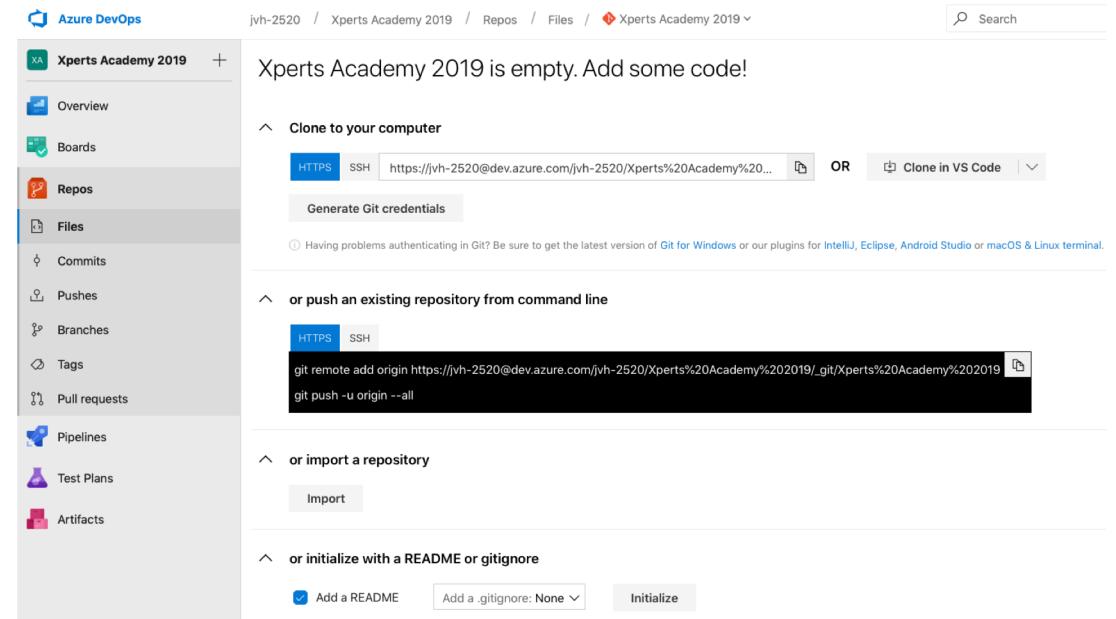
Xperts Academy 2019

Visibility can be public or private.



Import the ARM Template project

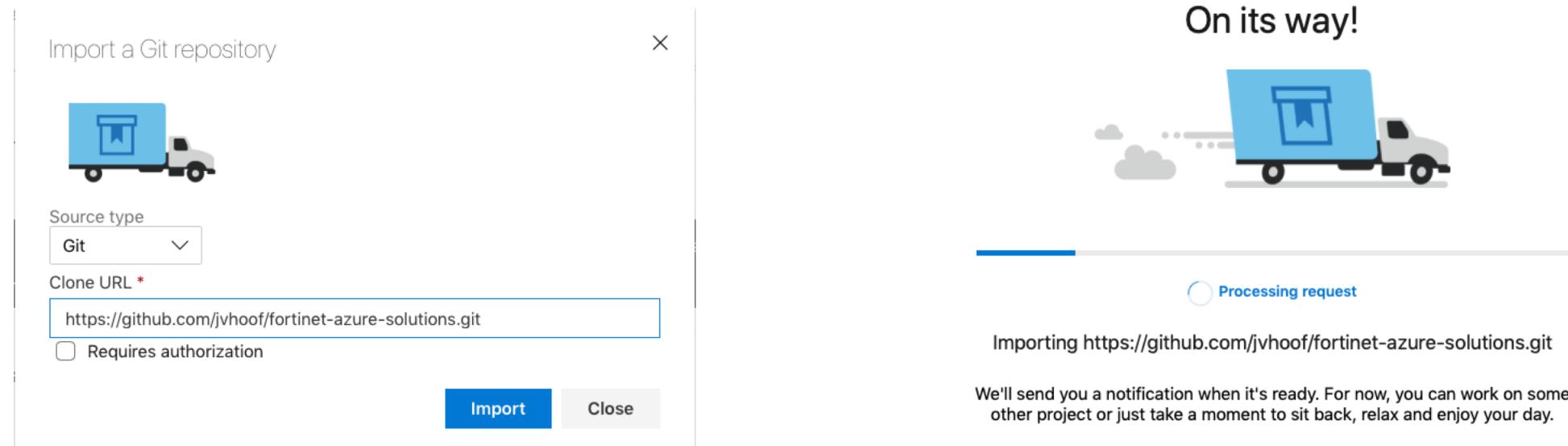
Go to Repos and select import



Import the ARM Template project

Import from the following Github project:

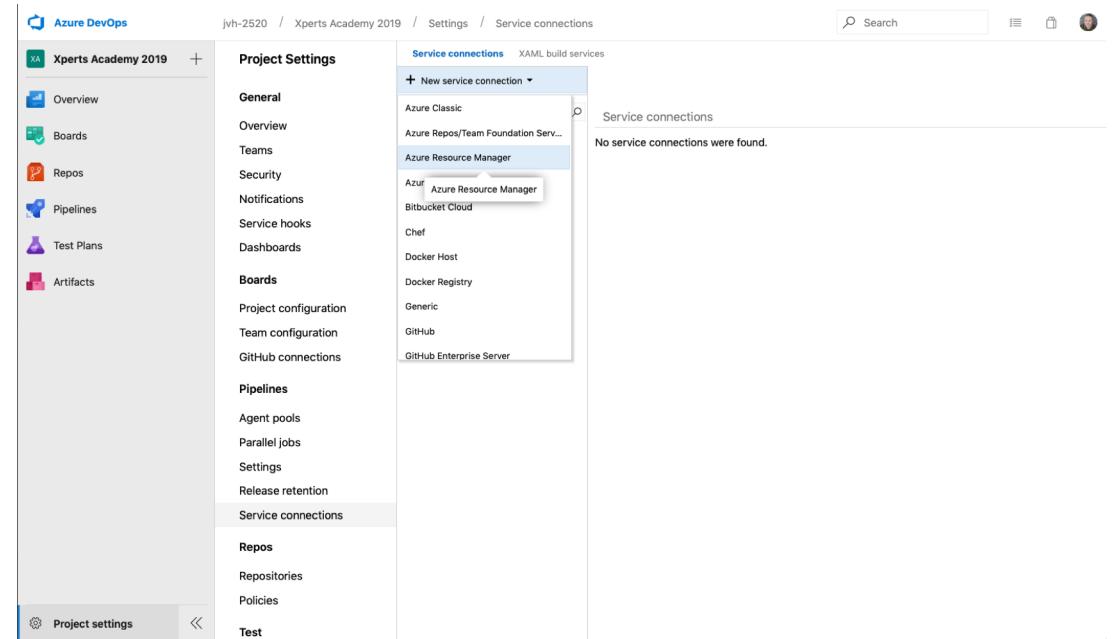
<https://github.com/jvhoof/Fortinet-azure-solutions.git>



Connect Azure DevOps with Azure

Open the **Project settings** and look for **Service Connections**

Create a **New Service Connections** for Azure Resource Manager



Connect Azure DevOps with Azure

Use the automatic way if you use the same credentials for Azure and Azure DevOps

Give the Connection the following name in one word and no quote's:
“AzureSubscription”

Add an Azure Resource Manager service connection

Service Principal Authentication Managed Identity Authentication

Connection name

Scope level

Subscription

Resource Group

Subscriptions listed are from Azure Cloud

A new Azure service principal will be created and assigned with "Contributor" role, having access to all resources within the subscription. Optionally, you can select the Resource Group to which you want to limit access.

If your subscription is not listed above, or your organization is not backed by Azure Active Directory, or to specify an existing service principal, [use the full version of the service connection dialog](#).

Allow all pipelines to use this connection.

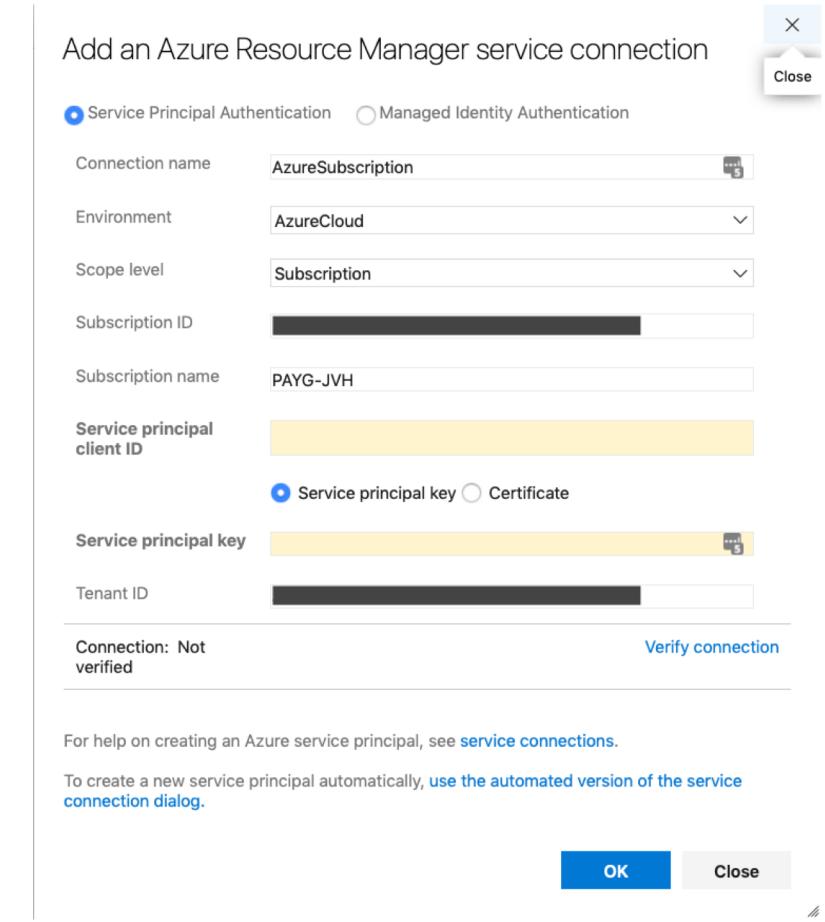
OK **Close**

Connect Azure DevOps with Azure

Create a **service principal** or use **managed identity authentication** in case you have a different account

<https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal>

<https://docs.microsoft.com/en-us/powershell/azure/create-azure-service-principal-azuresps?view=azps-2.3.2>



Create your first build pipeline

Open the **Pipeline** menu in the blade

Click on the ‘Create Pipeline’ button



Create your first Pipeline

Automate your build and release processes using our wizard, and go from code to cloud-hosted within minutes.

[Create Pipeline](#)

Create your first build pipeline

Select the Git repository you have the code imported to

New pipeline

✓ Connect **Select** Configure Review

Select a repository

Filter by keywords Xperts Academy 2019

- Xperts Academy 2019

Connect Select Configure Review

New pipeline

Where is your code?

- Azure Repos Git YAML
Free private Git repositories, pull requests, and code search
- Bitbucket Cloud YAML
Hosted by Atlassian
- Github YAML
Home to the world's largest community of developers
- Github Enterprise Server YAML
The self-hosted version of GitHub Enterprise
- Other Git
Any generic Git repository
- Subversion
Centralized version control by Apache

Use the [classic editor](#) to create a pipeline without YAML.

Create your first build pipeline

Azure Pipelines like most CI/CD pipelines have a configuration YAML file per project they can read in with the required steps

Select one of the three YAML pipelines Azure DevOps offers you

And run your pipeline!

The screenshot shows the 'Configure' step of creating a new pipeline. The 'Existing Azure Pipelines YAML file' option is selected, and a modal window titled 'Select an existing YAML file' is open. The modal shows the 'Branch' dropdown set to 'master' and the 'Path' dropdown set to '/FortiGate/Active-Passive-ELB-ILB/azure-pipelines.yml'. A small note at the bottom left of the modal says 'Xperts Academy 2019'.

The result

Azure DevOps interface showing the Pipelines page for the project "Xperts Academy 2019". The left sidebar lists various project management and development tools. The main area displays a recent pipeline run titled "#20190701.1 Update link", which was manually triggered from the master branch. The run was completed 4 minutes ago and took less than a second.

| Description | Stages |
|---|---------------|
| #20190701.1 Update link Manually triggered from master c7da0f7 | 4m ago <1s |

The result

A screenshot of the Azure DevOps Pipelines interface. The left sidebar shows the project navigation with 'Pipelines' selected. The main area displays the 'Jobs in run #20190701.2' for the 'Xperts Academy 2019' pipeline. A specific job is shown with its details:

- Job information:**
 - Pool: Hosted VS2017
 - Agent: Hosted Agent
 - Started: Just now
 - The agent request is already running or has already completed.
- Job preparation parameters:**
 - Initialize job (1s)
 - Checkout (8s)
- Azure PowerShell script: FilePath**
 - Import-Module -Name C:\Modules\azurerm_6.7.0\AzureRM\6.7.0\AzureRM.psd1 -Global
 - Publish Test Results **\TEST-*.xml
 - Post-job: Checkout

Demo and what's next

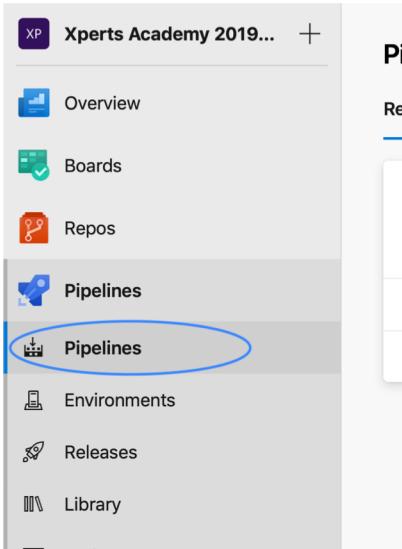
- Continuous integration:
 - Verify that the trigger is working when you commit a change.
- Editor: Visual Studio Code is a very good editor to work on git code
 - <https://code.visualstudio.com/>
 - Import code in Visual Studio Code
 - <https://www.azuredevopslabs.com/labs/azuredevops/git/>
 - Commit an update and verify the build running

Infrastructure as Code

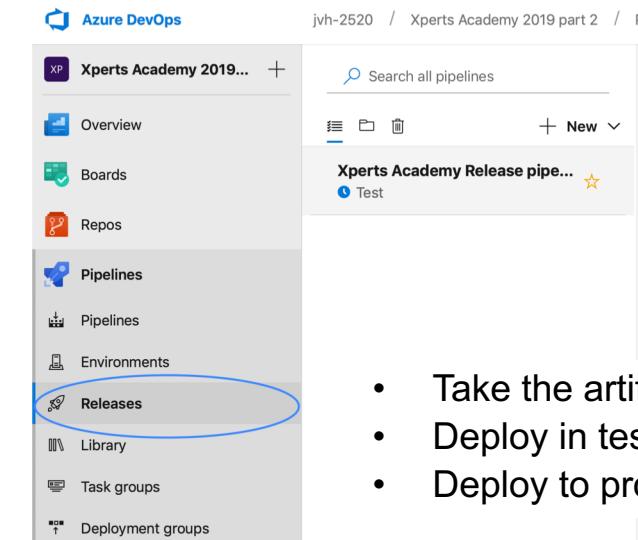
LAB 2 : Build and release

Infrastructure as Code

LAB 2



BUILD pipeline



RELEASE pipeline

- Package template and additional information as an artifact
- Validate the template

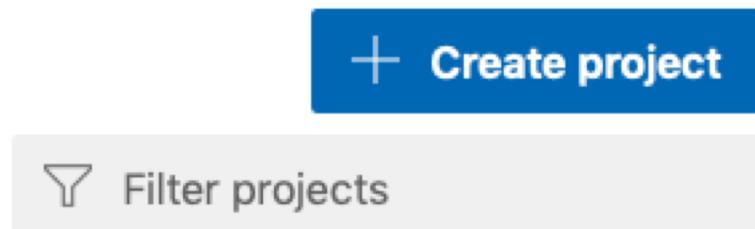
- Take the artifact package
- Deploy in test, validate
- Deploy to production

Build and release

Create a new project in the main organization page:

Xperts Academy 2019 Part 2

Visibility can be public or private.



Create new project

Project name *

 ✓

Description

Visibility

Public
Anyone on the internet can view the project.
Certain features like TFVC are not supported.

Private
Only people you give access to will be able to view this project.

Advanced

Version control ?

 ▼

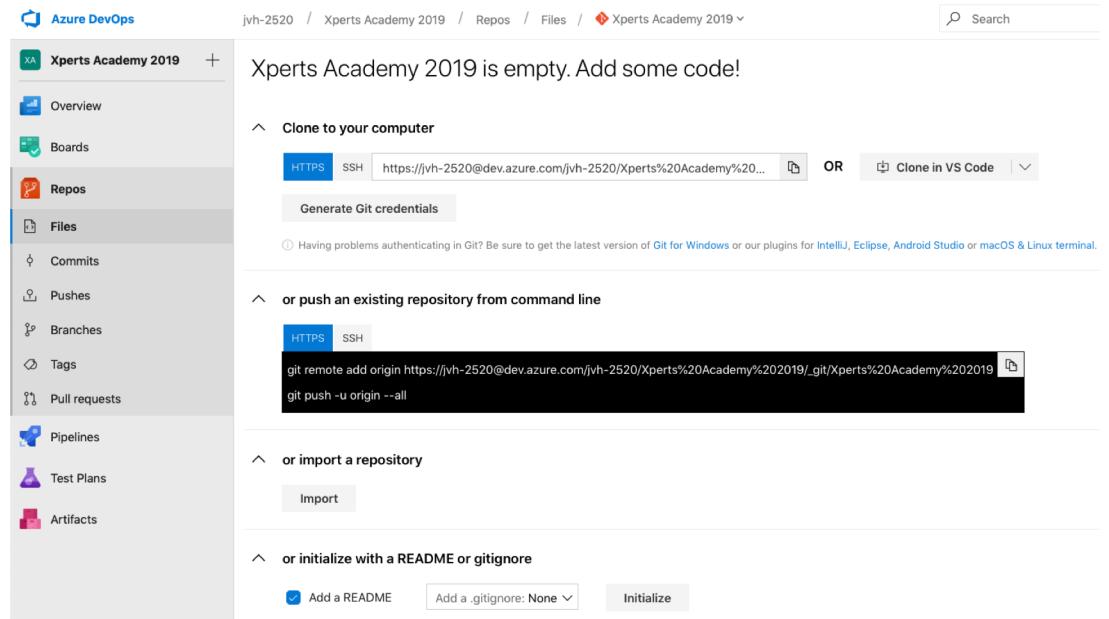
Work item process ?

 ▼

Cancel Create

Import the ARM Template project

Go to Repos and select import



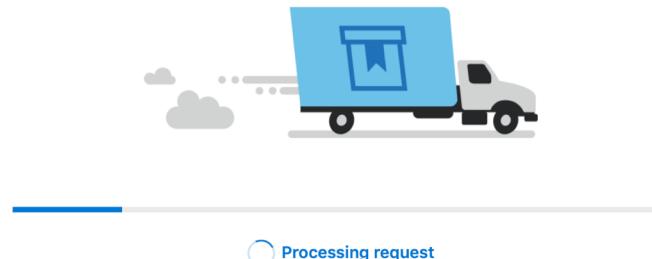
Import the ARM Template project

Import from the following Github project:

<https://github.com/jvhoof/xperts-academy-2019.git>

The screenshot shows the Azure DevOps interface for a project named 'Xperts Academy 2019 part 2'. The 'Repos' tab is selected in the sidebar. A modal window titled 'Import a Git repository' is open, showing the URL 'https://github.com/jvhoof/xperts-academy-2019'. The 'Source type' dropdown is set to 'Git'. Below the URL input field, there is a checkbox for 'Requires authorization' which is unchecked. At the bottom of the modal are 'Import' and 'Close' buttons.

On its way!



Processing request

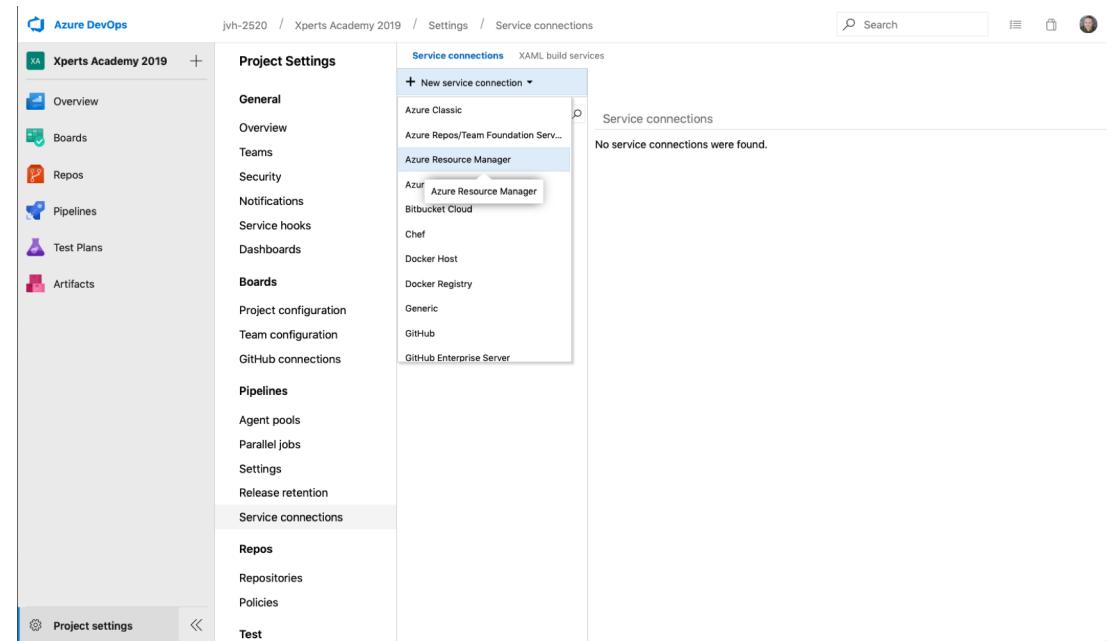
Importing https://github.com/jvhoof/xperts-academy-2019

We'll send you a notification when it's ready. For now, you can work on some other project or just take a moment to sit back, relax and enjoy your day.

Connect Azure DevOps with Azure

Open the **Project settings** and look for **Service Connections**

Create a **New Service Connections** for Azure Resource Manager



Connect Azure DevOps with Azure

Use the automatic way if you use the same credentials for Azure and Azure DevOps

Give the Connection the following name in one word and no quote's:
“AzureSubscription”

Add an Azure Resource Manager service connection

Service Principal Authentication Managed Identity Authentication

Connection name

Scope level

Subscription

Resource Group

Subscriptions listed are from Azure Cloud

A new Azure service principal will be created and assigned with "Contributor" role, having access to all resources within the subscription. Optionally, you can select the Resource Group to which you want to limit access.

If your subscription is not listed above, or your organization is not backed by Azure Active Directory, or to specify an existing service principal, [use the full version of the service connection dialog](#).

Allow all pipelines to use this connection.

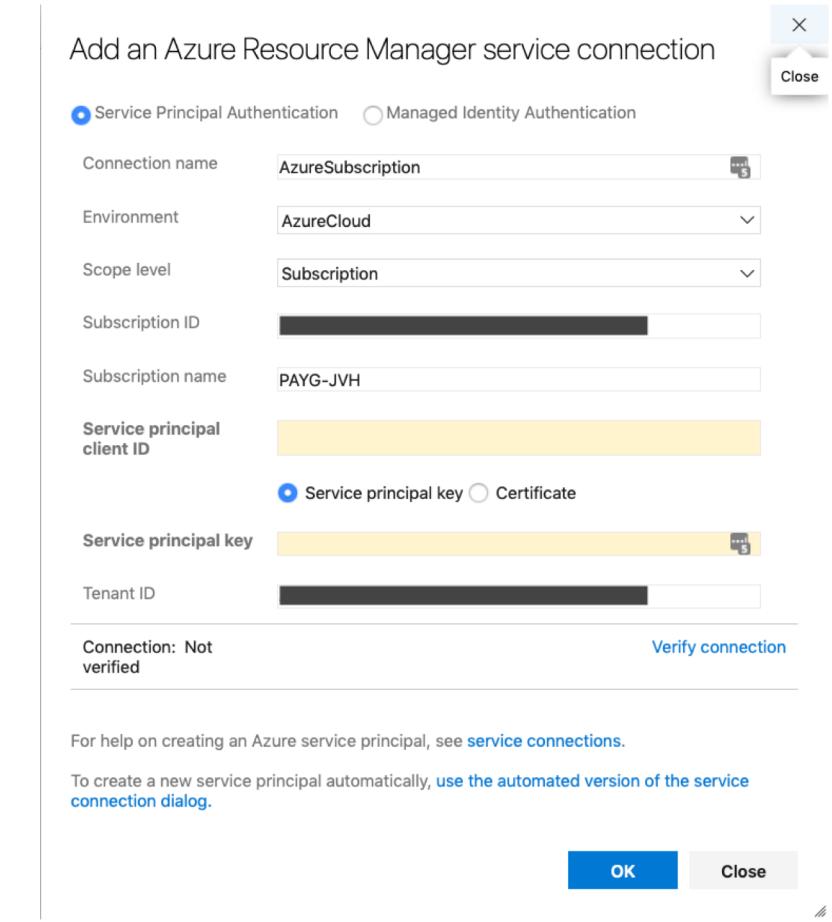
OK **Close**

Connect Azure DevOps with Azure

Create a **service principal** or use **managed identity authentication** in case you have a different account

<https://docs.microsoft.com/en-us/azure/active-directory/develop/howto-create-service-principal-portal>

<https://docs.microsoft.com/en-us/powershell/azure/create-azure-service-principal-azuresps?view=azps-2.3.2>



Create a new build pipeline

Create a new build pipeline using the classic editor

The screenshot shows the Azure DevOps Pipelines interface. On the left, there is a sidebar with various project management and development tools: Overview, Boards, Repos, Pipelines, Environments, Releases, Library, Task groups, Deployment groups, Test Plans, and Artifacts. The 'Pipelines' option is highlighted with a red arrow. The main area is titled 'Where is your code?' and lists several integration options:

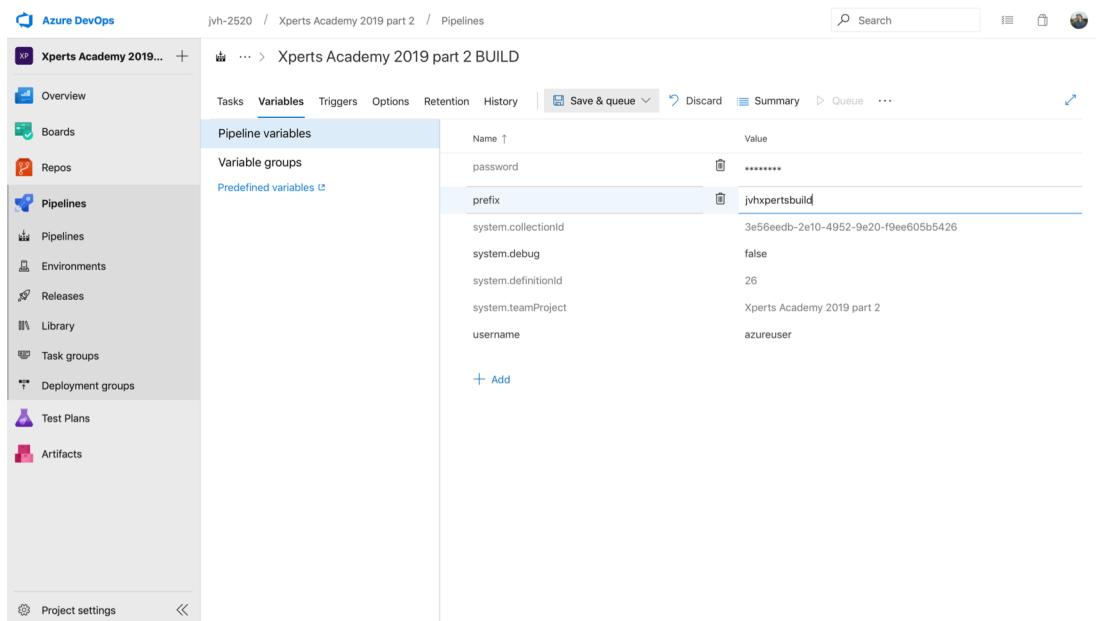
- Azure Repos Git (YAML)
- Bitbucket Cloud (YAML)
- Github (YAML)
- Github Enterprise Server (YAML)
- Other Git
- Subversion

Below the list, a note says: "Use the classic editor to create a pipeline without YAML."

Create a new build pipeline

Specify 3 variables and make sure to mask the password:

- username
- password
- prefix: max 12 characters, numbers and lowercase letters



The screenshot shows the 'Variables' tab in the Azure DevOps Pipelines interface. The 'Pipeline variables' section lists the following variables:

| Name | Value |
|---------------------|--------------------------------------|
| password | ***** |
| prefix | jvhxpertsbuild |
| system.collectionId | 3e56eedb-2e10-4952-9e20-f9ee605b5426 |
| system.debug | false |
| system.definitionId | 26 |
| system.teamProject | Xperts Academy 2019 part 2 |
| username | azureuser |

Create a new build pipeline

Add the first task: Azure Resource Group deployment

The screenshot shows the Azure DevOps Pipeline editor interface. At the top, it displays the path: "... > Xperts Academy 2019 part 2 BUILD". Below this is the pipeline configuration area with tabs for Tasks, Variables, Triggers, Options, Retention, and History. The Tasks tab is selected. The pipeline consists of three main stages:

- Get sources**: Set to Xperts Academy 2019 part 2, master branch.
- Agent job 1**: Set to Run on agent.
- Azure Deployment:Create Or Update Resource Group ac...**: A task with a warning message: "Some settings need attention".

To the right of the pipeline, there is a sidebar titled "Add tasks" with a search bar containing "Res". Below the search bar, several tasks are listed:

- Azure Resource Manager deployment output parser**: Converts ARM template deployment output to Azure DevOps pipeline variables version 1.0.0.
- Publish Test Results**: Publishes test results to Azure Pipelines.
- Azure resource group deployment**: Deploy an Azure Resource Manager (ARM) template to a resource group and manage virtual machines. This task is highlighted with a blue background and has an "Add" button.
- Publish code coverage results**: Publish Cobertura or JaCoCo code coverage results from a build.

Create a new build pipeline

And configure it with the following variables:

- Resource Group
- Location
- Template
- Template variables
- Deployment mode: validation only

The screenshot shows the Azure DevOps Pipeline editor interface. The pipeline is named "Xperts Academy 2019 part 2 BUILD". It contains two tasks: "Get sources" and "Agent job 1". The "Agent job 1" task contains an "Azure Deployment" task. The "Azure Deployment" task is configured to "Create Or Update Resource Group" action, with the "Resource group" set to "JVH-XPERTS-ACADEMY-BUILD" and "Location" set to "West Europe".

Create a new build pipeline

And configure it with the following variables:

- Resource Group
- Location
- Template
- Template variables
- Deployment mode: validation only

Template ^

Template location * ⓘ

Linked artifact ▾

Template * ⓘ

azuredeploy.json ▾ ...

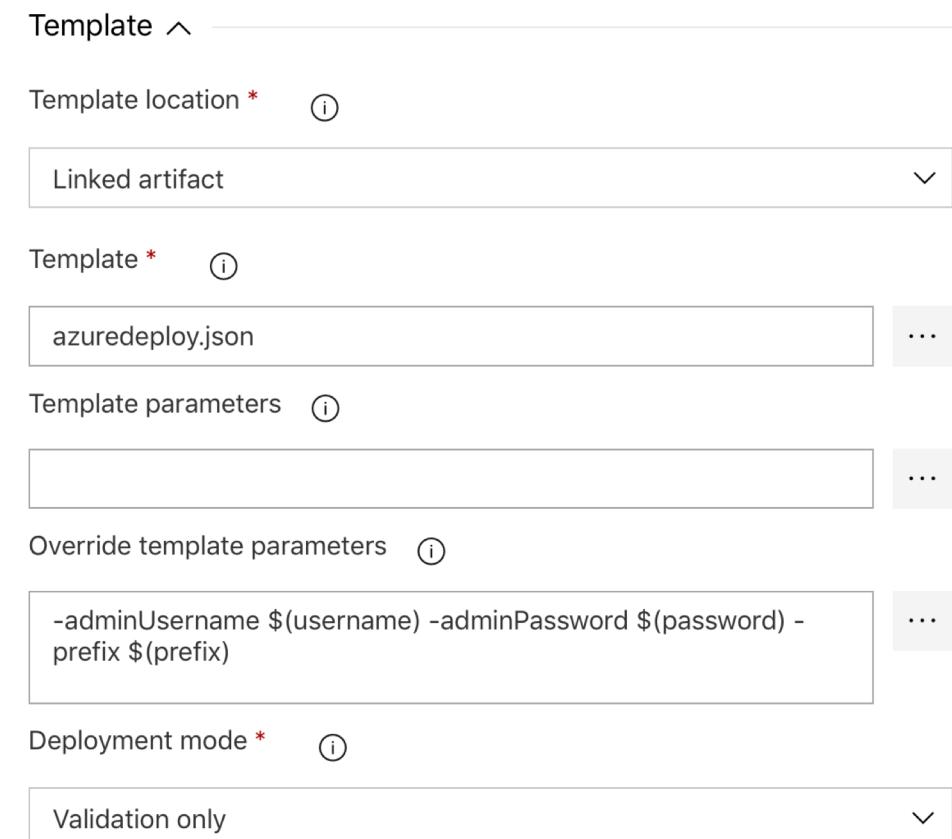
Template parameters ⓘ

Override template parameters ⓘ

-adminUsername \$(username) -adminPassword \$(password) -
prefix \$(prefix) ▾ ...

Deployment mode * ⓘ

Validation only ▾



Create a new build pipeline

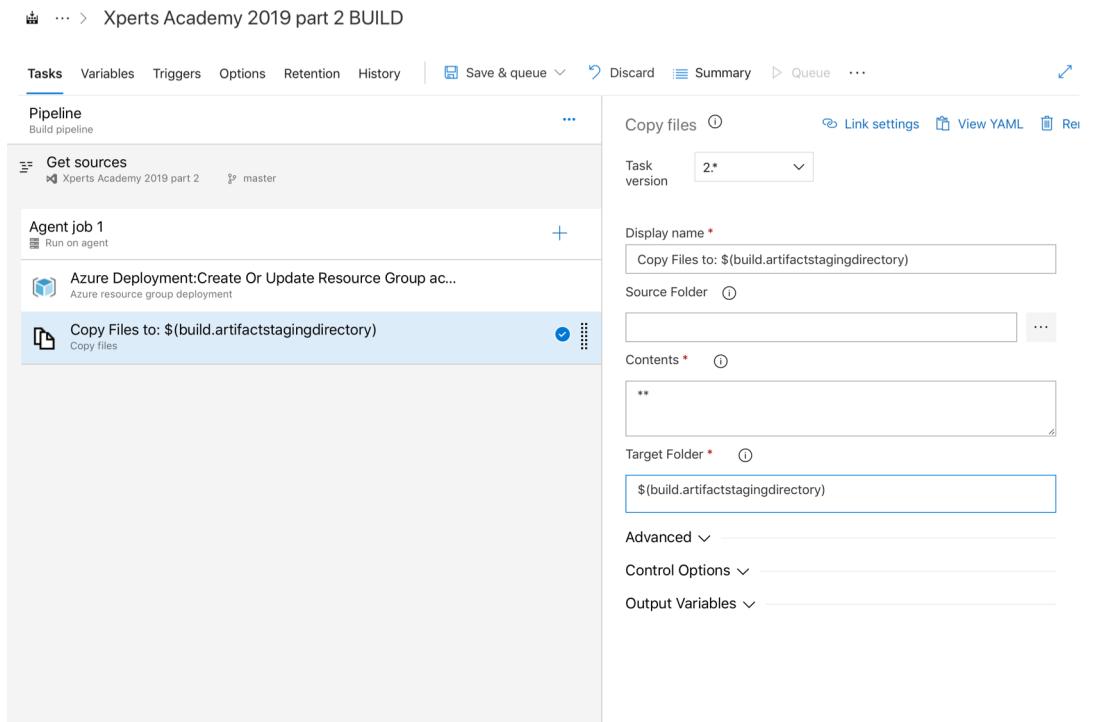
Add a copy task to copy the deployment files to the artifacts directory

The screenshot shows the Azure DevOps Pipeline editor interface. At the top, there's a breadcrumb navigation: Pipeline > Xperts Academy 2019 part 2 BUILD. Below the navigation, there are tabs for Tasks, Variables, Triggers, Options, Retention, and History. On the right, there are buttons for Save & queue, Discard, Summary, Queue, and more. A search bar on the right is set to 'copy'. The main area displays a 'Pipeline' build pipeline with three tasks: 'Get sources', 'Agent job 1', and 'Copy Files to:'. The 'Copy Files to:' task has a note: 'Some settings need attention'. To the right of the pipeline, a sidebar titled 'Add tasks' lists several options under the 'Copy' category, with 'Copy files' highlighted. Other listed tasks include 'Copy files over SSH', 'Windows machine file copy', and 'Azure file copy'. At the bottom of the sidebar, there are 'Add' and 'Learn more' buttons.

Create a new build pipeline

Add a copy task to copy the deployment files to the artifacts directory.

- Source: empty
- Target folder:
\$(build.artifactstagingdirectory)



Create a new build pipeline

Add a publish build artifacts task

The screenshot shows the Azure Pipelines interface for a build pipeline named "Xperts Academy 2019 part 2 BUILD". The pipeline consists of a single agent job named "Agent job 1" which runs on an agent. This job contains several tasks: "Get sources" (using Xperts Academy 2019 part 2 master branch), "Azure Deployment:Create Or Update Resource Group ac...", "Copy Files to: \${build.artifactstagingdirectory}" (Copy files), and "Publish Artifact: drop" (Publish build artifacts). The "Publish Artifact: drop" task is currently selected, indicated by a blue border around its card. To the right of the pipeline editor, there is a sidebar titled "Add tasks" with a search bar containing "Artif". Below the search bar, a list of available tasks is shown:

- Publish pipeline artifact**: Publish (upload) a file or directory as a named artifact for the current run.
- Download pipeline artifact**: Download build and pipeline artifacts.
- Download build artifacts**: Download files that were saved as artifacts of a completed build.
- Download artifacts from file share**: Download artifacts from a file share, like \\share\drop.
- Publish build artifacts**: Publish build artifacts to Azure Pipelines or a Windows file share. This task is highlighted with a blue background and has an "Add" button.
- Jenkins download artifacts**: Download artifacts produced by a Jenkins job.
- npm**: Manage npm packages.

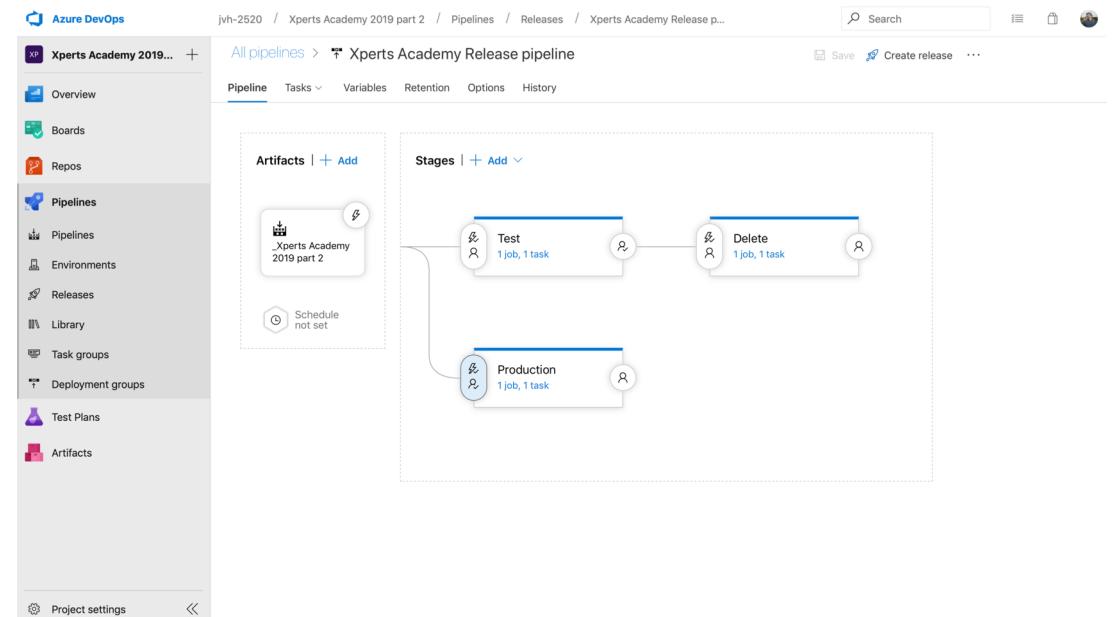
Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals



Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals

The screenshot shows the Azure DevOps Pipeline editor for the 'Xperts Academy Release pipeline'. The pipeline has a single stage named 'Test'. Under this stage, there is an 'Agent job' task with a single 'Azure Deployment' step. The 'Azure Deployment' step is configured to 'Create Or Update Resource Group' action on the 'JVH-XPERTS-ACADEMY-RELEASE-TEST' resource group in 'West Europe' location. The 'Task version' is set to 2*. The pipeline is currently at Task version 2.

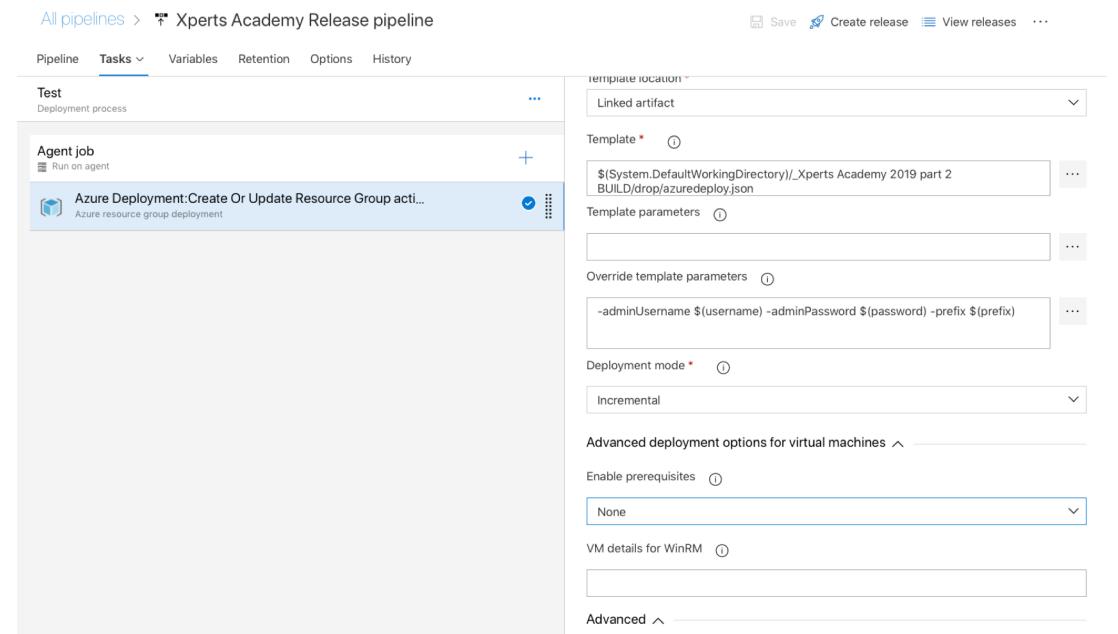
Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals



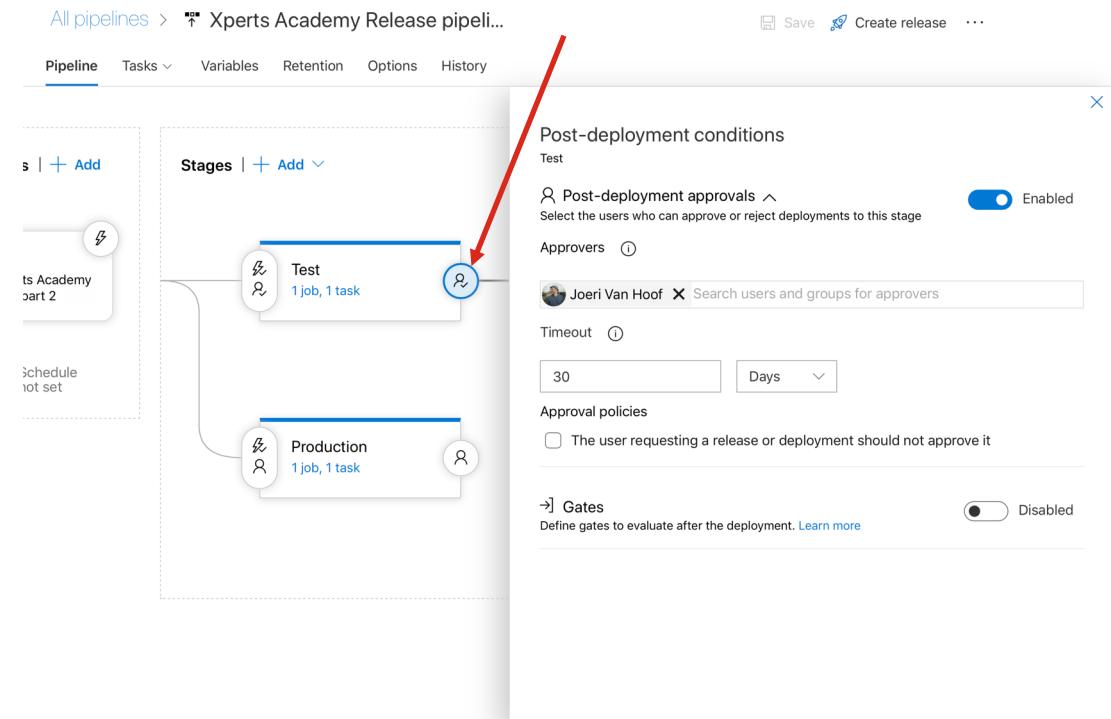
Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals



Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals

The screenshot shows the Azure DevOps Pipeline editor for the 'Xperts Academy Release pipeline'. The pipeline has a single step named 'Delete' under the 'Deployment process' tab. This step contains an 'Agent job' task, which in turn contains an 'Azure Deployment:DeleteRG' action. The action is configured to delete the resource group 'JVH-XPERTS-ACADEMY-RELEASE-TEST'. Other configuration options like 'Task version', 'Display name', and 'Azure Details' (subscription and action) are also visible.

Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals

The screenshot shows the Azure DevOps Pipeline editor for the 'Xperts Academy Release pipeline'. The pipeline has a single stage named 'Production'. Under 'Production', there is an 'Agent job' step. Within the 'Agent job' step, there is one task: 'Azure Deployment:Create Or Update Resource Group action...'. The task configuration shows the following details:

- Azure resource group deployment**: Task version 2*
- Display name**: Azure Deployment:Create Or Update Resource Group action on JVH-XPERTS-ACADEM...
- Azure Details**:
 - Azure subscription: AzureSubscription (Scoped to subscription 'PAYG-JVH')
 - Action: Create or update resource group
 - Resource group: JVH-XPERTS-ACADEMY-RELEASE-2
 - Location: West Europe
- Template**: Template location

Create a new release pipeline

Create a test state

Establish post-deployment approvals

Verify the test stage is deleted

Deploy production after approvals

All pipelines > Xperts Academy Release pipeline

Pipeline Tasks Variables Retention Options History

Production Deployment process

Agent job Run on agent

Azure Deployment: Create Or Update Resource Group action... Azure resource group deployment

Template ^

Template location *

Linked artifact

Template * ⓘ

\$(System.DefaultWorkingDirectory)/_Xperts Academy 2019 part 2

Template parameters ⓘ

-adminUsername \$(username) -adminPassword \$(password) -prefix \$(prefix) -fortigateLicense="\$fortigateLicense"

Deployment mode *

In Advanced deployment options for virtual machines

Advanced deployment options for virtual machines ^

Enable prerequisites ⓘ

None

VM details for WinRM ⓘ

Infrastructure as Code

Step 3 : license file injection

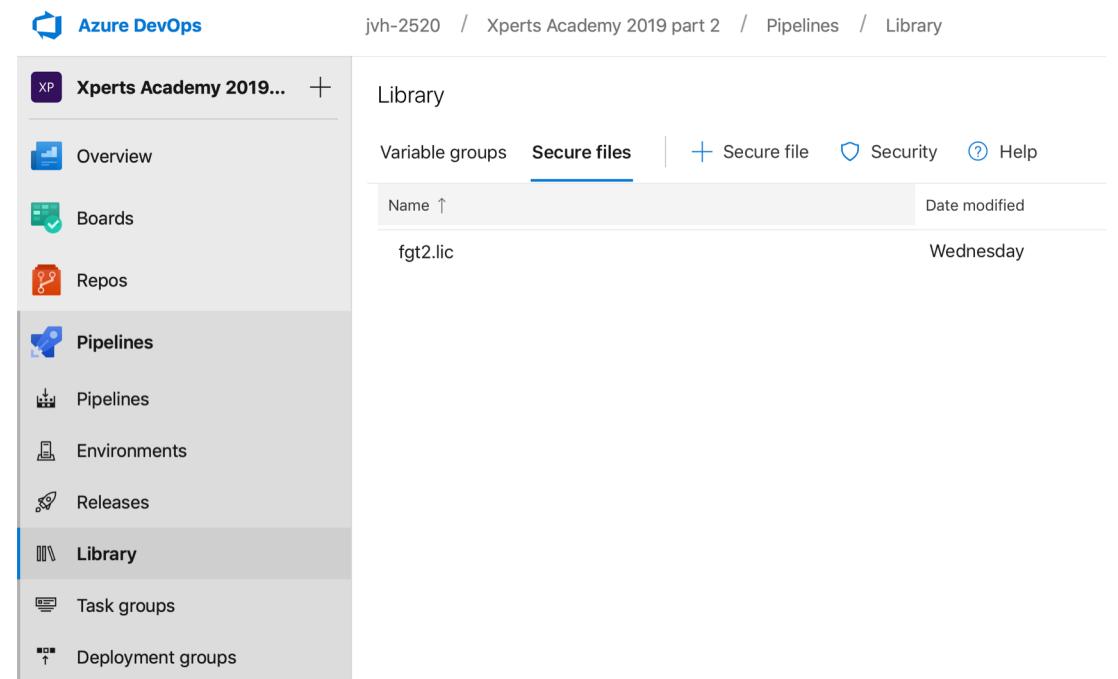
Create a new build pipeline

Upload a license to the secure file section

Create a build pipeline

Create a task to download the secure file to the agent

Create a task to run via Azure CLI the deployment using the provided deploy.sh script.



The screenshot shows the Azure DevOps interface with the following details:

- Header:** Azure DevOps, jvh-2520 / Xperts Academy 2019 part 2 / Pipelines / Library
- Sidebar:** Overview, Boards, Repos, Pipelines, Pipelines, Environments, Releases, **Library** (selected), Task groups, Deployment groups.
- Content Area:**
 - Secure files tab:** Selected. Shows a table with one item:

| Name ↑ | Date modified |
|----------|---------------|
| fgt2.lic | Wednesday |
 - Buttons:** + Secure file, Security, Help.

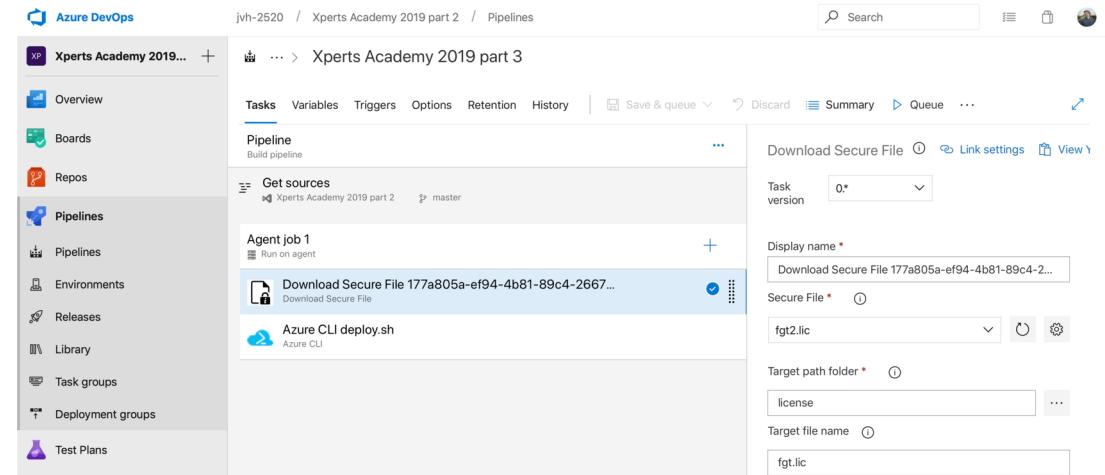
Create a new build pipeline

Upload a license to the secure file section

Create a build pipeline

Create a task to download the secure file to the agent

Create a task to run via Azure CLI the deployment using the provided deploy.sh script.



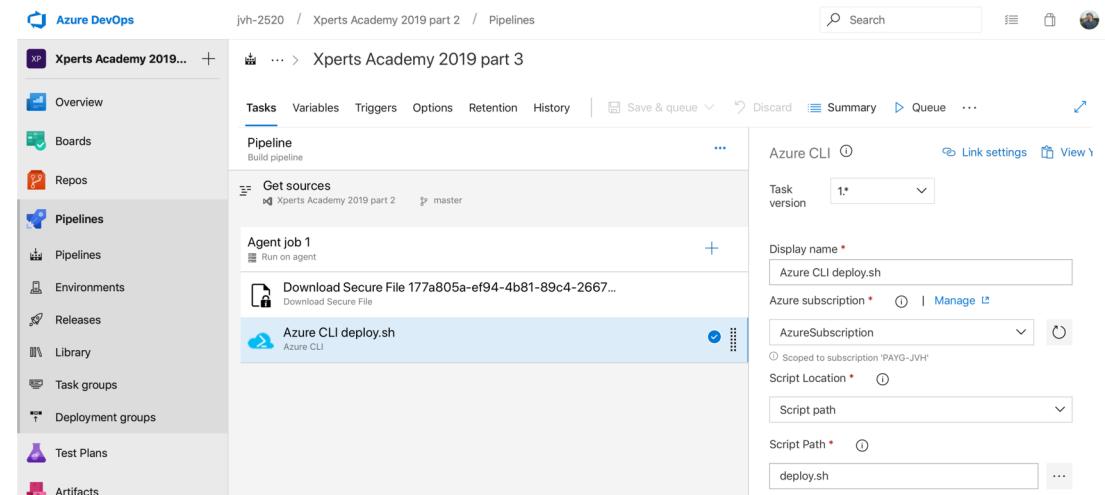
Create a new build pipeline

Upload a license to the secure file section

Create a build pipeline

Create a task to download the secure file to the agent

Create a task to run via Azure CLI the deployment using the provided deploy.sh script.



FORTINET[®]
NSE **ACADEMY**
XPERTS